AD-A102 162 NAVAL RESEARCH LAB WASHINGTON DC F/G 9/2
IMPLEMENTING BRENT'S MULTIPLE-PRECISION PACKAGE ON THE TEXAS IN--ETC(U)
JUL 81 A R MILLER, E VEGH
UNCLASSIFIED NRL-MR-4582 NL

END
DATE
FILMED
8 81
DTIC

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NRL Memorandum Report 4582	2. GOVT ACCESSION NO. AD-A102 162	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) IMPLEMENTING BRENT'S MULTIPLE-PRECISION PACKAGE ON THE TEXAS INSTRUMENTS ADVANCED SCIENTIFIC COMPUTER		5. TYPE OF REPORT & PERIOD COVERED Final report 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) A. R. Miller and E. Vegh		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Research Laboratory Washington, DC 20375		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 64211N; W0454-AA; 53-0661-A-0; 71756-7215
11. CONTROLLING OFFICE NAME AND ADDRESS Office of the Under Secretary of Defense Office of Combat Support		12. REPORT DATE July 24, 1981 13. NUMBER OF PAGES 12
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer software	Elementary functions	Gamma function
Multiple-precision computation	Special functions	Exponential integrals
Mathematical functions	Bessel functions	Special constants

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This note describes how to use Brent's multiple-precision arithmetic package newly made available for the ASC. This note also enumerates some of the concomitant mathematical functions, constants and software. In addition, a comparison of computation times for the basic arithmetic operations is made between this package and the existing ASC multiple-precision arithmetic software.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

CONTENTS

IMPLEMENTING BRENT'S MULTIPLE-PRECISION PACKAGE ON THE TEXAS INSTRUMENTS ADVANCED SCIENTIFIC COMPUTER

USING BRENT'S MULTIPLE-PRECISION PACKAGE ON THE ASC

This note will enable an ASC user to easily and quickly use MP, a multiple-precision arithmetic package written by Richard P. Brent of the Australian National University [1]. This note is intended as an introductory guide for using MP on the Texas Instruments (TI) ASC, and is not meant to replace the documentation for MP. Documentation for MP is catalogued at USERCAT/LIB/UNSUP/EXTDOC; additional documentation and source listing reside at USERCAT/LIB/UNSUP/EXTSCR. The object library (FX compilation) for MP is catalogued at USERCAT/LIB/UNSUP/EXTOBJ and contains 100 subroutines. Reference [1] is recommended for those who would like to make a deeper study of the mathematics of the MP package.

The following lists the minimum JSL for a typical batch job:

```
/ JOB....

/ ASG OBJ, USERCAT/LIB/UNSUP/EXTOBJ, USE = SHR

/ FTN

   fortran (main) program and subroutines (if any)

/ LNK LSPACE = XXXXX

LIBRARY OBJ

/ FXQT

/ EOJ
```

The following non-executable statements must be included in the calling program:

```
COMMON   B, T, M, LUN, MXR, R

INTEGER  B, T, R
```

R must be dimensioned and B, T, M, LUN, MXR must be defined.

i. For best efficiency and space utilization set B=16384.

ii. Error messages are printed on logical unit LUN, where LUN is set by the user; normally LUN=6.

iii. If N is the "equivalent" number of floating decimal places desired by the user, then T is the smallest integer such that

$$T \geq \max \left\{ 1+7(\log_{10}B)^{-1},\ 1+N(\log_{10}B)^{-1} \right\}.$$

For B=16384,

$$T \geq \max \left\{ 3,\ 1 + .24N \right\}.$$

Manuscript submitted June 5, 1981.

iv. Exponents can lie in the closed interval [-M,M], where M is set by the user. M must be greater than T and less than or equal to the integer part of 0.25 $(2^{31} - 1)$ = 536,870,911.

v. The value of MXR is obtained as follows: For each subroutine in the MP package is associated a parameter called SPACE, which is a linear function of T; i.e., SPACE has the form SPACE = $\alpha T + \beta$, where α and β are integers. Suppose that n subroutines belonging to the MP package are used. Then, for each of these routines we have:

$$\text{SPACE}_j = \alpha_j T + \beta_j, \quad 1 \le j \le n.$$

Suppose that the value of integer T as defined above has been chosen, i.e., $T = T_o$. Then MXR is defined by

$$\max_{1 \le j \le n} \left\{ \text{SPACE}_j \right\} = \max_{1 \le j \le n} \left\{ \alpha_j T_o + \beta_j \right\}.$$

vi. R must be dimensioned in the calling program; its dimension is MXR. This array is used as work space for the collection of MP routines used.

We note that the above parameters in common may be varied dynamically subject to the restrictions of the MP package and ASC FORTRAN. Further, we note that MP uses "T-digit floating point numbers" which are stored in integer arrays of dimension T+2 (see documentation and example programs).

It is important to note that the subroutine MPSET must not be used as it will produce integer overflow.*

TABLE OF BASIC ARITHMETIC ROUTINES AND SAMPLE PROGRAMS

There are 100 routines in the MP package. Some basic routines are [1, pp 70 - 74 or catalogued documentation previously mentioned]:

MPADD Adds two MP numbers, SPACE = T + 4

CALL MPADD (X, Y, Z) means Z = X + Y

---

* MPSET causes integer overflow when computer wordlength is less than 48 bits. Brent suggests two alternatives for MPSET; a rewriting of this software or setting the parameters B, T, M, LUN and MXR without calling MPSET. The latter has been chosen since it provides greater flexibility. MPSET was not removed from the MP package because it is mentioned (not used) elsewhere in the documentation.

MPSUB Subtracts one MP number from another, SPACE = T + 4

CALL MPSUB (X, Y, Z) means Z = X - Y

MPMUL Multiplies two MP numbers, SPACE = T + 4

CALL MPMUL (X, Y, Z) means Z = X * Y

MPDIV Divides two MP numbers, SPACE = 4T + 10

CALL MPDIV (X, Y, Z) means Z = X/Y

MPIN Converts fixed-point number read under A1 format to multiple precision, SPACE = 3T + 11

MPOUT Converts multiple-precision to a form suitable for printing under A1 format (corresponds to F or I formats), SPACE = 3T + 11

Appendix 1 gives a detailed program illustrating how these arithmetic routines may be used on the ASC. Program output also provided.

An example program TSTMP using MP subroutines is also given in Appendix 2 of this note; the output is provided. This example program is an ASC adaptation and extension of the example program supplied with the MP package. The program TSTMP uses (implicitly and explicitly) 49 of the 99 usable MP subroutines. The times for computations are included.

MATHEMATICAL FUNCTIONS, CONSTANTS AND SOFTWARE

The capabilities of MP include the following [1, p. 58-59]:

i. conversion of integer, real, and double-precision numbers to multiple-precision format and vice versa;

ii. multiplication and division of multiple-precision (mp) numbers by small integers;

iii. addition, subtraction, multiplication, and division of mp numbers;

iv. powers and roots of mp numbers;

v. elementary functions of mp numbers: arcsin, arctan, cos, cosh, exp, ln, sin, sinh, tan, tanh;

vi. some special functions: Bessel functions of the first kind, error and complementary error functions, exponential and logarithmic integrals, Dawson's integral, and gamma function;

vii. constants: Bernoulli numbers, π, Euler's constant, and the Riemann zeta function for positive integer arguments;

viii. fixed and floating-point decimal output and free-field decimal input of mp numbers;

ix. integer and fractional parts of mp numbers;

x. routines for error handling, testing, and debugging;

xi. miscellaneous: comparison of mp numbers, storing, packing and unpacking mp numbers, etc.

COMPUTATION TIME COMPARISONS

Computation times for the basic arithmetic operations of the MP package and for the TI package [2] are given in Tables 1 and 2. Table 3 clearly shows the superiority of MP over the Texas Instruments Extended Precision Package [2] with respect to each of the arithmetic operations.

IMPROVING MP ON THE ASC

Although the speed of the MP package is substantial (see timings listed in example output), the speed of the package could be increased by optimizing, for the ASC, the basic addition, multiplication, and division routines:

MPNZR, MPMLP, MPDIVI, MPADD2, MPADD3, MPMUL2.

TABLE 1

33 significant figures	+	-	X	÷
MP package	3,910	4,090	9,210	77,430
TI package	28,600	42,100	1,370,000	290,000

The entries of this table give the time in clocks (1 clock = 8×10^{-8} second) for operations on two multiple-precision numbers with 33 significant figures. TI package refers to reference [2].

TABLE 2

70 significant figures	+	-	X	÷
MP package	5,570	5,780	14,200	109,000
TI package	52,400	81,000	5,820,000	873,000

The entries of this table give the time in clocks for operations on two multiple-precision numbers with 70 significant figures.

TABLE 3

r	+	-	X	÷
33 significant figures	7.3	10.3	149	3.75
70 significant figures	9.4	14	410	8

The entries of this table give the ratio r of TI Extended Precision Package to MP Package for each operation performed on ASC.

CONCLUSION

Brent's multiple-precision arithmetic package is now available on the ASC. It not only performs the basic arithmetic operations up to orders of magnitude faster than what is now available on the ASC, but it is also supported by well-documented software of functions and constants of great interest in scientific computations.

ACKNOWLEDGMENT

We would like to thank Dr. Benjamin Lepson of the Management Information Division for recognizing the potential of Brent's work and for acquiring the MP package for the Research Computation Center in 1979.

We should also like to thank Jon Wilson for reading this paper and for his comments and suggestions.

REFERENCES

1. R. P. Brent, A Fortran Multiple-Precision Arithmetic Package. ACM Trans. Math. Software 4,1 (March 78), 57-81.

2. Extended Precision Subroutines for the Texas Instruments ASC, TI, Inc. publication 929947-1, 1 May 1976.

APPENDIX I

```
          SOURCE LISTING        ASC FAST FORTRAN COMPILER      RELEASE FTFX0529.P294/80              0001

ISN            STATEMENT        CP OPTIONS = (M,X)             DATE = 05/04/81(81.124)    TIME = 14:43:04

           C   LET PI=3.142..., E=2.718... THIS PROGRAM COMPUTES TO AT LEAST 21
           C   SIGNIFICANT FIGURES PI+E, PI-E, PI*E, PI/E.
           C
0001           COMMON B,T,M,LUN,MXR,R
0002           IMPLICIT INTEGER(A-Z)
           C
           C   SET BASE B
0003           DATA B/16384/
           C
           C   SET EXPONENT M
0004           DATA M/40/
           C
           C   SET OUTPUT UNIT LUN
0005           DATA LUN/6/
           C
           C   COMPUTATIONS WILL BE DONE TO 25 SIGNIFICANT FIGURES. THEREFORE
           C   SET T=7.
0006           DATA T/7/
           C
           C   MPPI REQUIRES 3T+8 WORDS, MPEXP REQUIRES 4T+10 WORDS,
           C   MPADD,MPSUB,MPMUL,MPDIV REQUIRE T+4 WORDS,
           C   MPOUT REQUIRES 3T+11 WORDS;
           C   THEREFORE SET MXR=38.
0007           DATA MXR/38/
           C
           C   ARRAY FOR WORK SPACE R IS DIMENSIONED BY MXR.
0008           DIMENSION R(38)
           C
           C   VARIABLES REQUIRE T+2 WORDS; THEREFORE:
0009           DIMENSION PI(9), E(9)
0010           DIMENSION R1(9), R2(9), R3(9), R4(9)
0011           DIMENSION ONE(9)
           C
0012           CALL RSSTOP
           C
           C   CONVERT DOUBLE-PRECISION '1' TO MULTIPLE-PRECISION '1'=ONE.
0013           CALL MPCDM(1.D0,ONE)
           C
           C   COMPUTE E
0014           CALL MPEXP(ONE,E)
           C   COMPUTE PI
0015           CALL MPPI(PI)
           C
           C   COMPUTE PI+E, PI-E, PI*E, PI/E RESPECTIVELY.
           C   RESULTS ARE THE RI,I=1,2,3,4.
0016           CALL MPADD(PI,E,R1)
0017           CALL MPSUB(PI,E,R2)
0018           CALL MPMUL(PI,E,R3)
0019           CALL MPDIV(PI,E,R4)
           C
```

```
         SOURCE LISTING          ASC FAST FORTRAN COMPILER       RELEASE FTFX0529.P294/80           0002

CSN         STATEMENT            CP OPTIONS = (M,X)           DATE = 05/04/81(81.124)     TIME = 14:43:04

       C
       C     CONVERT MULTIPLE-PRECISION RESULTS R1,R2,R3,R4 TO A FORM SUITABLE
       C     FOR PRINTING UNDER A1 FORMAT.
0020         DIMENSION OUT1(25), OUT2(25), OUT3(25), OUT4(25)
       C
0021         CALL MPOUT(R1,OUT1,25,20)
0022         CALL MPOUT(R2,OUT2,25,20)
0023         CALL MPOUT(R3,OUT3,25,20)
0024         CALL MPOUT(R4,OUT4,25,20)
       C
       C     PRINT RESULTS ON UNIT LUN.
0025         WRITE(LUN,11) OUT1
0026         WRITE(LUN,12) OUT2
0027         WRITE(LUN,13) OUT3
0028         WRITE(LUN,14) OUT4
0029      11 FORMAT(5X,"PI+E=",25A1,/)
0030      12 FORMAT(5X,"PI-E=",25A1,/)
0031      13 FORMAT(5X,"PI*E=",25A1,/)
0032      14 FORMAT(5X,"PI/E=",25A1,/)
0033         END


PI+E=   5.85987448204883847382

PI-E=   0.42331082513074800310

PI*E=   8.53973422267356706546

PI/E=   1.15572734979092171791
```

APPENDIX II

```
          SOURCE LISTING           ASC FAST FORTRAN COMPILER    RELEASE FTFX0529.P294/80           0001

ISN        STATEMENT               OP OPTIONS = (M,X)           DATE = 03/19/81(81.078)    TIME = 13:30:51

0001       PROGRAM TSTMP
0002       COMMON B,T,M,LUN,MXR,R
0003       INTEGER B,T,R(130)
0004       INTEGER T1, T2, CLOCK
       C
0005       DATA B/ 16384/
0006       DATA LUN/6/
0007       DATA M/100/
0008       DATA T/30/
0009       DATA MXR/130/
       C
       C
0010       INTEGER PI(32)
0011       INTEGER X(32)
0012       INTEGER C(110)
       C
0013       CALL RESTOP
0014       T1=CLOCK(T1)
       C
       C   COMPUTE PI
       C
0015       CALL MPPI(PI)
0016       T2=CLOCK(T2)
0017       TIME=(T2-T1)* 80.0E-9
0018       CALL MPOUT(PI,C,110,100)
0019       WRITE(LUN,10) B,T,C
0020       WRITE(6,1) TIME
0021       T1=CLOCK(T1)
       C
       C   COMPUTE EXP(PI*SQRT(163/9))
       C
0022       CALL MPQPWR(163,9,1,2,X)
0023       CALL MPMUL(X,PI,X)
0024       CALL MPEXP(X,X)
0025       T2=CLOCK(T2)
0026       TIME=(T2-T1)* 80.0E-9
0027       CALL MPOUT(X,C,110,100)
0028       WRITE(LUN,20) C
0029       WRITE(6,1) TIME
0030       T1=CLOCK(T1)
       C
       C   COMPUTE EXP(PI*SQRT(163))
       C
0031       CALL MPPWR(X,3,X)
0032       T2=CLOCK(T2)
0033       TIME=(T2-T1)* 80.0E-9
0034       CALL MPOUT(X,C,110,90)
0035       WRITE(LUN,30) C
0036       WRITE(6,1) TIME
       C
```

```
TSTMP          SOURCE LISTING          ASC FAST FORTRAN COMPILER       RELEASE FTFX0529.P294/80            0002
   ISN           STATEMENT             CP OPTIONS = (M,X)              DATE = 03/19/81(81.078)    TIME = 13:30:51

            C      COMPUTE SIN(70)
            C
   0037            INTEGER  X1,Y1
   0038            INTEGER D
   0039            DIMENSION D(40)
   0040            I=7
   0041            WRITE(LUN,11) B,T
   0042            DIMENSION X1( 9)
   0043            DIMENSION Y1( 9)
            C
   0044            CALL MPCIM(70,X1)
   0045            T1=CLOCK(T1)
   0046            CALL MPSIN(X1,Y1)
   0047            T2=CLOCK(T2)
   0048            TIME=(T2-T1)* 80.0E-9
   0049            CALL MPOUT(Y1,D,40,23)
   0050            WRITE(6,40) D
   0051            WRITE(6,1) TIME
            C
            C
            C      COMPUTE BESSEL FUNCTION ORDER 0 EVAL AT 0.1
            C
   0052            INTEGER XB,YB
   0053            DIMENSION XB(7), YB(7)
   0054            T=5
   0055            WRITE(6,11) B,T
   0056            CALL MPCDM(.1D0,XB)
   0057            T1=CLOCK(T1)
   0058            CALL MPBES2(XB,0,YB)
   0059            T2=CLOCK(T2)
   0060            TIME=(T2-T1)*80.0E-9
   0061            CALL MPOUT(YB,D,40,15)
   0062            WRITE(6,50) D
   0063            WRITE(6,1) TIME
   0064            STOP
            C
   0065          1 FORMAT(/,'  TIME = ' ,F15.3,'  SECONDS',/)
   0066         10 FORMAT(32H1EXAMPLE OF MP PACKAGE,   BASE =, I9,
                 $ 12H,   DIGITS =, I4 /// 11H PI TO 1000D //
                 $ 11X, 60A1 / 21X, 50A1)
   0067         11 FORMAT(32H EXAMPLE OF MP PACKAGE,   BASE =, I9,
                 $ 12H,   DIGITS =, I4 ,// )
   0068         20 FORMAT (/ 28H EXP(PI*SQRT(163/9)) TO 1000D //
                 $          11X, 60A1 / 21X, 50A1)
   0069         30 FORMAT (/ 25H EXP(PI*SQRT(163)) TO 90D //
                 $          1X, 70A1 / 21X, 40A1)
   0070         40 FORMAT(//,5X,' SIN(70) CORRECT TO 23 SIG FIG',//,1X,40A1,//)
   0071         50 FORMAT(//,'   BESSEL FUNCTION ORDER 0 EVAL AT 0.1 CORRECT',/,
                 $4X,'TO 15 SIG FIG',40A1)
   0072            END
```

```
EXAMPLE OF MP PACKAGE,   BASE =    16384,   DIGITS =  30

PI TO 1000

                    3.14159265358979323846264338327950288419716939937510
                      58209749445923078164062862089986280348253421170680
 TIME =             0.189  SECONDS

EXP(PI*SQRT(163/9)) TO 1000

              640320.00000000060486373504921653947174181881653947577148
                      57603665918194652218258286942536340815822646477590
 TIME =             0.424  SECONDS

EXP(PI*SQRT(163)) TO 900

 262537412640768743.99999999999925007259719818568887935385633733699086
                      2707537410378210647910118607312951181346
 TIME =             0.015  SECONDS
EXAMPLE OF MP PACKAGE,   BASE =    16384,   DIGITS =   7


     SIN(70) CORRECT TO 23 SIG FIG

              0.773890681557889097787 33


 TIME =             0.063  SECONDS
EXAMPLE OF MP PACKAGE,   BASE =    16384,   DIGITS =   5


 BESSEL FUNCTION ORDER 0 EVAL AT 0.1 CORRECT
  TO 15 SIG FIG                          0.997501562066040

 TIME =             0.027  SECONDS
```

DATE
ILME